Objective 3 homework

Consider the following annotated program:

$\{n = n_0 \wedge n_0 > 0\}$

```
f = 1
while n > 0 do
  f = f * n
  n = n - 1
end
```

$\{f = n_0!\}$

The correctness of this program can be proved with the following invariant:

$f \cdot n! = n_0! \wedge n0$

For example, in the programming language ruby this invariant can be placed in the program and evaluated to verify that it holds throughout the execution as follows:

```
f = 1
while n > 0 do
  raise "bad invariant" unless f*fact(n)==fact(n0) && n>=0
  f = f * n
  n = n - 1
end
```

Problem 1

Discover an invariant suitable for proving the correctness of the following program. Code the program in you favored language. Include in the program an annotation that *evaluates* the invariant (possibly in several places) to verify that the invariant holds and consequently the program is correct.

$\{n \geqslant 0\}$

```
s = 1
i = 0
while i < n do
  s = s * 2
  i = i + 1
end
```

$\{s = 2^n\}$

Problem 2

Redo problem 1 for the following program:

$\{n \geqslant 0\}$
```
sq = 0
i = 0
while i < n do
  sq = sq + 2*i + 1
  i = i + 1
end
```
$\{sq = n^2\}$

Problem 3

Redo problem 1 for the following program, where `a` is an array and `a.size` is the number of its elements:

$\{\mathit{true}\}$
```
tot = 0
i = 0
while i < a.size do
  tot = tot + a[i]
  i = i + 1
end
```
$\{tot = \sum_{k=0}^{a.size-1} a[k]\}$